Exact Ramsey Theory: Green-Tao numbers and SAT

Oliver Kullmann

Computer Science Department
Swansea University
O.Kullmann@Swansea.ac.uk
http://cs.swan.ac.uk/~csoliver

Abstract. We consider the links between Ramsey theory in the integers, based on van der Waerden's theorem, and (boolean, CNF) SAT solving. We aim at using the problems from exact Ramsey theory, concerned with computing Ramsey-type numbers, as a rich source of test problems, where especially methods for solving hard problems can be developed. We start our investigations here by reviewing the known *van der Waerden numbers*, and we discuss directions in the parameter space where possibly the growth of van der Waerden numbers $\mathrm{vdw}_m(k_1, \ldots, k_m)$ is only polynomial (this is important for obtaining feasible problem instances). We introduce *transversal extensions* as a natural way of constructing mixed parameter tuples $(k_1, \ldots, k_m)$ for van-der-Waerden-like numbers $\mathrm{N}(k_1, \ldots, k_m)$, and we show that the growth of the associated numbers is guaranteed to be linear. Based on Green-Tao's theorem ("the primes contain arbitrarily long arithmetic progressions") we introduce the *Green-Tao numbers* $\mathrm{grt}_m(k_1, \ldots, k_m)$, which in a sense combine the strict structure of van der Waerden problems with the (pseudo-)randomness of the distribution of prime numbers. Using standard SAT solvers (look-ahead, conflict-driven, and local search) we determine the basic values. It turns out that already for this single form of Ramsey-type problems, when considering the best-performing solvers a wide variety of solver types is covered. For $m > 2$ the problems are non-boolean, and we introduce the *generic translation scheme*, which offers an infinite variety of translations ("encodings") and covers the known methods. In most cases the special instance called *nested translation* proved to be far superior over its competitors (including the direct translation).

1 Introduction

The applicability of SAT solvers has made tremendous progress over the last 15 years; see the recent handbook [3]. We are concerned here with solving (concrete) combinatorial problems (see [33] for an overview). Especially we are concerned

with the computation of van-der-Waerden-like numbers, which is about colouring hypergraphs of arithmetic progressions.[1)]

An *arithmetic progression* of size $k \in \mathbb{N}_0$ in $\mathbb{N}$ is a set $P \subset \mathbb{N}$ of size k such that after ordering (in the natural order), two neighbours always have the same distance. So the arithmetic progressions of size $k > 1$ are the sets of the form $P = \{a + i \cdot d : i \in \{0, \dots, k-1\}\}$ for $a, d \in \mathbb{N}$. Van der Waerden's Theorem ([32]) shows that whenever the set $\mathbb{N}$ of natural numbers is partitioned into finitely many parts, some part must contain arithmetic progressions of arbitrary size. The finite version, which is equivalent to the above infinite version, says that for every progression size $k \in \mathbb{N}$ and every number $m \in \mathbb{N}$ of parts there exists some $n_0 \in \mathbb{N}$ such that for $n \geq n_0$ every partitioning of $\{1, \dots, n\}$ into m parts has some part which contains an arithmetic progression of size k. The smallest such n_0 is denoted by $\mathbf{vdw}_{\boldsymbol{m}}\boldsymbol{(k)}$, and is called a *vdW-number*. The subfield of Ramsey theory concerned with van der Waerden's theorem is for over 70 years now an active field of mathematics and combinatorics; for an elementary introduction see [23].

We are concerned here with *exact Ramsey theory*, that is, computing vdW-like numbers if possible, or otherwise producing (concrete) lower bounds. [6] introduced the application of SAT for computing vdW-numbers, showing that all known vdW-numbers (at that time) were rather easily computable with SAT solvers. With [14] yet SAT had its biggest success, computing the new (major) vdW-number $\mathrm{vdw}_2(6) = 1132$ (the problem of computing $\mathrm{vdw}_2(6)$ is mentioned in [23] as a difficult research problem). See [1,2] for the current state-of-the-art. Regarding lower bounds, the best lower bounds currently one finds in [11].[2)]

VdW-numbers for "core" parameter values (see Definition 2) grow rapidly, and thus only few are known (see Section 3). The first contribution of this article is the notion of a *transversal extension* (see Definition 2) of a parameter tuple, which allows to grow parameter tuples such that (only) linear growth of the associated vdW-numbers is guaranteed. The linear growth is proven in a general framework in Theorem 10, and applied to vdW-numbers in Corollary 11.

Next we introduce *Green-Tao numbers* ("GT-numbers"; see Definition 13), which are defined as the vdW-numbers but using the first n *prime numbers* instead of the first n natural numbers. The existence of these numbers is given by the celebrated Green-Tao Theorem ([9]). In Corollary 14 we show that also for GT-numbers transversal extension numbers grow only linearly. In the remainder of the article we are concerned with computing GT-numbers.

For binary parameter tuples ($m = 2$ above) the problems of computing vdW- or GT-numbers have a canonical translation to (boolean) SAT problems, while for $m > 2$ we still have a canonical translation into non-boolean SAT problems

[1)] This report is an extended version of [21].

[2)] see `http://www.st.ewi.tudelft.nl/sat/waerden.php` for updates

(as is the case in general for hypergraph colouring problems; see [19]), but for using standard (boolean) SAT solvers the problem of a boolean translation arises. In Section 4 we introduce the *generic translation scheme*, with seven natural instances (amongst them the well-known direct and logarithmic translations). As it turns out, in nearly all cases for all solver types the *weak nested translation* (introduced in [17]) performed far best, with the only exception that for relatively large numbers of colours the logarithmic translation was better.

For this (initial) phase of investigations into GT-numbers we just used "off-the-shelves" SAT solvers, also aiming at some form of basic understanding why which type of solver is best on certain parameter ranges. For over one year on average 10 processors were running, with a lot of manual interaction and adjustment to find the right solvers and translations, and to set the parameters (most basic the number of vertices), establishing the basic Green-Tao numbers. All generators and the details of the computations are available in the open-source research platform `OKlibrary` (see [16]).[3)] See Section 5 for the results of these computations. We conclude this article by a discussion of interesting research directions in Section 6.

2 A few notions and notations

We use $\mathbb{N}_0 = \mathbb{Z}_{\geq 0}$ and $\mathbb{N} = \mathbb{N}_0 \setminus \{0\}$. A *finite hypergraph* G is a pair $G = (V, E)$ where V is a finite set and $E \subseteq \mathbb{P}(V)$ (that is, E is set of subsets of V); we use $V(G) := V$ and $E(G) := E$. An *m-colouring* of a hypergraph G is a map $f : V(G) \to \{1, \dots, m\}$ such that no hyperedge is monochromatic, that is, for every $H \in E(G)$ there are $v, w \in H$ with $f(v) \neq f(w)$. Regarding (boolean) clause-sets, complementation of boolean variables v is denoted by $\overline{v}$, (boolean) clauses are finite and clash-free sets of (boolean) literals, and (boolean) clause-sets are finite sets of (boolean) clauses.

3 The theorem of Green-Tao, and Green-Tao numbers

The numbers $\mathrm{vdw}_m(k)$ introduced in Section 1 are "diagonal vdW-numbers", while we consider also the "non-diagonal" or *mixed vdW-numbers*, which are defined as follows.

Definition 1. *A **parameter tuple** is an element of $\mathbb{N}_{\geq 2}^m$ for some $m \in \mathbb{N}$ which is monotonically non-decreasing (that is, sorted in non-decreasing order). For a parameter tuple $(k_1, \dots, k_m)$ the **vdW-number** $\boldsymbol{\mathrm{vdw}_m(k_1, \dots, k_m)}$ is the smallest $n_0 \in \mathbb{N}$ such that for every $n \geq n_0$ and every $f : \{1, \dots, n\} \to \{1, \dots, m\}$ there exists some "colour" (or "part") $i \in \{1, \dots, m\}$ such that $f^{-1}(i)$ contains an arithmetic progression of size k_i.*

---
3) `http://www.ok-sat-library.org`

In a systematic study of parameter tuples and their operations, one likely should drop the sorting condition, and call our parameter tuples "sorted", however in this report we only consider sorted parameter tuples.

Obviously we have $\mathrm{vdw}_m(k_1, \ldots, k_m) \leq \mathrm{vdw}_m(\max(k_1, \ldots, k_m))$ (note that the right-hand side denotes a diagonal vdW-number), and thus also the mixed vdW-numbers exist (are always finite). The most up-to-date collection of precise (mixed) vdW-numbers one finds in [1,2].[4] For the sake of completeness we state the numbers here, but for references we refer to [1,2]. We introduce the following organisation of the parameter space.

Definition 2. *A parameter tuple is **trivial** if all entries are equal to* 2*, otherwise it is **non-trivial**. A **simple** parameter tuple has length* 1*, otherwise it is **non-simple**. A parameter tuple is a **core tuple** if it is non-simple and if all entries are greater than or equal to* 3*. A parameter tuple t is a **(transversal) extension** of a parameter tuple t' if t can be obtained from t' by adding entries equal to* 2 *to the front of t'. A transversal extension of a simple parameter tuple is called an **extended simple tuple** or a **transversal tuple**, while an transversal extension of a core tuple is called an **extended core tuple**. Finally a parameter tuple is **diagonal**, if it is constant (all entries are equal), while otherwise it is **non-diagonal** or **mixed**.*

*Accordingly we speak of (and are interested in) **trivial vdW-numbers**, **simple vdW-numbers**, **core vdW-numbers**, **transversal vdW-numbers**, **extended core vdW-numbers**, and **diagonal vdW-numbers**.*

The trivial vdW-numbers are $\mathrm{vdw}_m(2) = m+1$, while the simple vdW-numbers are given by $\mathrm{vdw}_1(k) = k$. The known core vdW-numbers are as follows.

1. 24 binary core vdW-numbers $\mathrm{vdw}_2(a, b)$ are known:

a \ b	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
3	9	18	22	32	46	58	77	97	114	135	160	186	218	238	279	312
4	-	35	55	73	109	146										
5	-	-	178	206												
6	-	-	-	1132												

2. 4 core vdW-numbers $\mathrm{vdw}_3(a, b, c)$ and one core vdW-number $\mathrm{vdw}_4(a, b, c, d)$ are known:

a, b \ c	3	4	5
3, 3	27	51	80
3, 4	-	89	

,

a, b, c \ d	3
3, 3, 3	76

.

[4] The (updated) online-version is `http://users.encs.concordia.ca/~ta_ahmed/vdw.html`.

A basic quest for this article is in what "directions" can one move through the parameter space while experiencing only polynomial growth? Regarding vdW-numbers we consider the conjecture (perhaps better called "question"), that extended core parameter tuples grow only polynomially in the extension length, where for parameter tuples a, b by $a; b$ we denote their concatenation:

Conjecture 3. For every (fixed) parameter tuple t of length m the map $k \in \mathbb{N} \mapsto \mathrm{vdw}_{m+1}(t; (k))$ is polynomially bounded in k (depending on t).

In other words, for the above tables and all similarly constructed tables growth in every row is polynomially bounded. The evidence for Conjecture 3 is as follows.

1. The case $t = (3)$, and more precisely $\mathrm{vdw}_2(3, k) \le k^2$, has been suggested in [5]. The numbers $\mathrm{vdw}_2(3, k)$ are known for $1 \le k \le 18$ (see above). Additionally, our experiments yield the following conjectured values (where using "$\ge x$" means that we believe that actually equality holds, while the lower bound "$> x - 1$" has been shown), further supporting the conjectured upper bound:[5] $\mathrm{vdw}_2(3, 19) \ge 349$, $\mathrm{vdw}_2(3, 20) \ge 389$, $\mathrm{vdw}_2(3, 21) \ge 416$.
2. Considering $t = (4)$, the numbers $\mathrm{vdw}_2(4, k)$ are known for $1 \le k \le 8$ (see above), while the bound $\mathrm{vdw}_2(4, 9) > 254$ is in [1]; we can improve this to $\mathrm{vdw}_2(4, 9) \ge 309$, and furthermore $\mathrm{vdw}_2(4, 10) > 328$. So going from $k = 8$ to $k = 9$ we see a rather big jump, however possibly from $k = 9$ to $k = 10$ only a small change might take place.
3. For general $t = (k_0)$ with $k_0 \ge 3$, in [5] the lower bound $\mathrm{vdw}_2(k_0, k) \ge k^{k_0 - 1 - \log(\log(k))}$ for sufficiently large k has been shown. It seems consistent with current knowledge that we could have $\mathrm{vdw}_2(k_0, k) \le k^{k_0 - 1}$ for all $k, k_0 \ge 1$.

A contribution of this article is the systematic consideration of transversal extensions as defined in Definition 2. The known $33 + 10 + 1 + 6 = 50$ extended core vdW-numbers are as follows (again, for references see [1]); transversal vdW-numbers are presented in Section A.1; see Subsection 3.3 for general remarks.

1. Extending $(3, k)$ by m 2's, i.e., the numbers $\mathrm{vdw}_{m+2}(2, \ldots, 2, 3, k)$:

m \ k	3	4	5	6	7	8	9	10	11	12	13
0	9	18	22	32	46	58	77	97	114	135	160
1	14	21	32	40	55	72	90	108	129	150	171
2	17	25	43	48	65	83	99	119			
3	20	29	44	56	72	88					
4	21	33	50	60							
5	24	36									
6	25										
7	28										

[5] All lower bounds are obtained by local-search algorithms from the Ubcsat-suite (see [31]), and all data is available through the `OKlibrary`.

2. Extending $(4,k)$ resp. $(5,k)$ by m 2's, i.e., numbers $\mathrm{vdw}_{m+2}(2,\ldots,2,4,k)$ resp. $\mathrm{vdw}_{m+2}(2,\ldots,2,5,k)$:

$m \backslash k$	4	5	6	7	8
0	35	55	73	109	146
1	40	71	83	119	
2	53	75	93		
3	54	79			
4	56				

,

$m \backslash k$	5
0	178
1	180

.

3. Extending $(3,3,k)$ by m 2's, i.e., numbers $\mathrm{vdw}_{m+3}(2,\ldots,2,3,3,k)$:

$m \backslash k$	3	4	5
0	27	51	80
1	40	60	86
2	41	63	
3	42		

Note that by Conjecture 3 we would have in every row only polynomial growth. Now in Corollary 11 we will prove that in every column we have *linear growth*, where actually the factor can be made as close to 1 as one wishes, when only m is big enough.

3.1 A general perspective on Ramsey theory

We consider a sequence $(G_n)_{n\in\mathbb{N}}$ of finite hypergraphs, where we assume that we have $V(G_n) \subseteq V(G_{n+1})$ and $E(G_n) \subseteq E(G_{n+1})$ for all n. Furthermore we assume $V(G_1) \neq \emptyset$ and $\forall\, n \in \mathbb{N} : \emptyset \notin E(G_n)$ for simplicity. Such a sequence of hypergraphs we call *nontrivial monotonic*. We consider the following questions:

(i) Does there exist some $n \in \mathbb{N}$ with $E(G_n) \neq \emptyset$?

(ii) Does for every $m \in \mathbb{N}$ exists some $n_0(m) \in \mathbb{N}$ such that for all $n \geq n_0$ the hypergraph G_n is not m-colourable? In this case we say that $(G_n)_{n\in\mathbb{N}}$ has the *Ramsey property*.

(iii) Does $\lim_{n\to\infty} \frac{\alpha(G_n)}{|V(n)|} = 0$ hold, where $\alpha(G)$ for a hypergraph G is the *independence number* of G, the maximum size of an independent vertex set (not containing any hyperedge)? In this case we say that $(G_n)_{n\in\mathbb{N}}$ has the *Szemerédi property*.

Clearly (ii) implies (i), while in turn (iii) implies (ii), since colouring a hypergraph G with m colours just means to partition $V(G)$ into at most m independent subsets. Considering the original vdW-problem, we have $G_n = \mathrm{ap}(k,n)$ for some fixed $k \in \mathbb{N}$, where $V(\mathrm{ap}(k,n)) = \{1,\ldots,n\}$, while $E(\mathrm{ap}(k,n))$ is the set

of arithmetic progressions of size k in $\{1, \dots, n\}$. Property (i) trivially holds, while property (ii) is van der Waerden's theorem. And property (iii) has been conjectured by Erdös and Turán in 1936 ([8]), and was finally proved by Szemerédi in his landmark paper [30] (for arbitrary k, one of the deepest results in combinatorics; for $k = 3$ it was proven in [26], for $k = 4$ in [29]).

Consider a nontrivial monotonic sequence $G = (G_n)_{n \in \mathbb{N}}$ of hypergraphs. The following definition generalises diagonal vdW-numbers, and introduces a form of "convergence rate" capturing the Szemerédi property.

Definition 4. *For $m \in \mathbb{N}$ let $\mathbf{N}_{\boldsymbol{m}}\boldsymbol{(G)} \in \mathbb{N} \cup \{+\infty\}$ be the infimum of $n \in \mathbb{N}$ such that G_n is not m-colourable. And for $q \in \mathbb{R}_{>0}$ let $\mathbf{cr}\boldsymbol{(G, q)} \in \mathbb{N} \cup \{+\infty\}$ be the infimum of $n \in \mathbb{N}$ such that for all $n' \geq n$ holds $\frac{\alpha(G_{n'})}{|V(G_{n'})|} < q$.*

Thus G has the Ramsey property iff for all $m \in \mathbb{N}$ we have $\mathrm{N}_m(G) < +\infty$, while G has the Szemerédi property iff for all $q \in]0, 1]$ we have $\mathrm{cr}(G, q) < +\infty$. Considering the sequence $(\mathrm{ap}(k, n))_{n \in \mathbb{N}}$ of vdW-hypergraphs of arithmetic progressions of size k we have $\mathrm{N}_m(\mathrm{ap}(k, -)) = \mathrm{vdw}_m(k)$. The following simple fact makes the above remark, that (iii) implies (ii), more precise.

Lemma 5. *For all $m \in \mathbb{N}$ we have $\mathrm{N}_m(G) \leq \mathrm{cr}(G, \frac{1}{m})$.*

We say that hypergraph sequences $G^1, \dots, G^m$ are *compatible* if for all n we have $V(G^1_n) = \dots = V(G^m_n)$. Generalising the notion of "diagonal vdW-like numbers" in Definition 4 and the notion of mixed vdW-numbers in Definition 1:

Definition 6. *Consider $m \in \mathbb{N}$ and compatible nontrivial monotonic hypergraph sequences $G^1, \dots, G^m$. Then $\mathbf{N}_{\boldsymbol{m}}\boldsymbol{(G^1, \dots, G^m)} \in \mathbb{N} \cup \{+\infty\}$ is defined as the infimum of $n \in \mathbb{N}$ such that for every m-colouring of $V(G^1_n)$ there exists some $i \in \{1, \dots, m\}$ such that some hyperedge of G^i_n is monochromatically i-coloured.*

Obviously we have $\mathrm{N}_m(G) = \mathrm{N}_m(G, \dots, G)$. Call $(G^1, \dots, G^m)$ *horizontally monotonic* if for all $n \in \mathbb{N}$ and all $1 \leq i \leq j \leq m$ every independent subset of G^i_n is also independent in G^j_n. In this case then $\mathrm{N}_m(G^1, \dots, G^m) \leq \mathrm{N}_m(G_m)$ holds. This captures the typical application of "mixed numbers" from Ramsey theory. Generalising the notion of "convergence rate" in Definition 4:

Definition 7. *Consider $m \in \mathbb{N}$ and compatible nontrivial monotonic hypergraph sequences $G^1, \dots, G^m$. For $q \in \mathbb{R}_{>0}$ let $\mathbf{cr}\boldsymbol{((G_1, \dots, G_m), q)} \in \mathbb{N} \cup \{+\infty\}$ be the infimum of $n \in \mathbb{N}$ such that for all $n' \geq n$ and for all m-tuples $(S_1, \dots, S_m)$ of (pairwise) disjoint independent subsets S_i of $G^i_{n'}$ we have $\frac{|S_1| + \dots + |S_m|}{n'} < q$.*

A few basic remarks:

1. $\mathrm{cr}(G, q) = \mathrm{cr}((G), q)$.

2. $\mathrm{cr}((G^1, \dots, G^m), q) \le \mathrm{cr}(G_m, \frac{q}{m})$ if $(G^1, \dots, G^m)$ is horizontally monotonic.
3. By definition we have for arbitrary compatible nontrivial monotonic hypergraph sequences that $\mathrm{N}_m(G^1, \dots, G^m) = \mathrm{cr}((G^1, \dots, G^m), 1)$. By Remark 2) this generalises Lemma 5.

For complete hypergraphs we can easily establish the Szemerédi property:

Lemma 8. *For $n, k \in \mathbb{N}$ let V_n^k be the hypergraph with vertex set $\{1, \dots, n\}$ and hyperedge set $\binom{\{1,\dots,n\}}{k}$. Now for natural numbers $k_1, \dots, k_m$ and $q > 0$ we have that $\mathrm{cr}((V_-^{k_1}, \dots, V_-^{k_m}), q)$ is the smallest $n > \frac{(\sum_{i=1}^m k_i) - m}{q}$. Especially we have $\mathrm{cr}((V^2, \dots, V^2), q) = \mathrm{cr}(V^{m+1}, q)$, which is the smallest $n > \frac{m}{q}$.*

By definition we get the following generalisation of Remark 2 to Definition 7:

Lemma 9. *For $m \in \mathbb{N}$ consider compatible nontrivial monotonic hypergraph sequences $G^1, \dots, G^m$, and consider $1 \le t < m$. Then for $p, q \in \mathbb{R}_{>0}$ we have*

$$\mathrm{cr}((G^1, \dots, G^m), p + q) \le \max\big(\mathrm{cr}((G^1, \dots, G^t), p),\ \mathrm{cr}((G^{t+1}, \dots, G^m), q)\big).$$

Lemma 8 and Lemma 9 (splitting $1 = \frac{1}{s} + (1 - \frac{1}{s})$) together yield the basic theoretical observation of this paper:

Theorem 10. *Consider $l \in \mathbb{N}$ and compatible nontrivial monotonic hypergraph sequences $G^1, \dots, G^l$. For $n \in \mathbb{N}$ let $V_n := V(G_n^1)$, and for $k \in \mathbb{N}$ let $Q_n^k := (V_n, \binom{V_n}{k})$, and thus $Q^k = (Q_n^k)_{n \in \mathbb{N}}$ is a nontrivial monotonic hypergraph sequence. For $x \in \mathbb{R}$ let $M(x) \in \mathbb{N} \cup \{+\infty\}$ be the infimum of $n \in \mathbb{N}$ such that we have $|V_n| > x$. Now for every $s \in \mathbb{R}$ with $s > 1$ and for every $m \in \mathbb{N}_0$ we have*

$$\mathrm{N}_{l+m}(G^1, \dots, G^l, Q^2, \dots, Q^2) = \mathrm{N}_{l+1}(G^1, \dots, G^l, Q^{m+1}) \le \max\Big(M(s \cdot m),\ \mathrm{cr}\,\big((G^1, \dots, G^l), 1 - \frac{1}{s}\big)\Big).$$

So the growth-rate of $m \mapsto \mathrm{N}_{l+m}(G^1, \dots, G^l, Q^2, \dots, Q^2)$ is linear for m large enough, where the factor can be made arbitrarily close to 1. Applied to vdW-numbers, using Szemerédi's theorem, we get the following application (the proof-idea here originated from Jan-Christoph Schlage-Puchta). As a special case of Definition 7 we use $\mathrm{cr}_{\mathrm{ap}}(t, q)$ for parameter tuples t, using the hypergraph sequences belonging to the progression sizes in t.

Corollary 11. *For a parameter tuple t of length $l \in \mathbb{N}$, for $m \in \mathbb{N}_0$ and for $s \in \mathbb{R}_{>1}$ we have $\mathrm{vdw}_{m+l}((2, \dots, 2); t) \le \max(s \cdot m + 1, \mathrm{cr}_{\mathrm{ap}}(t, 1 - \frac{1}{s}))$.*

Giving up on the factor, but now without unknown minimal value for m, we have the following variation on Corollary 11:

Lemma 12. *For a parameter tuple t of length $l \in \mathbb{N}$ and for $m \in \mathbb{N}_0$ we have* $\mathrm{vdw}_{m+l}((2,\ldots,2);t) \leq (m+1) \cdot \mathrm{vdw}_l(t)$.

Proof. Let $n := (m+1) \cdot \mathrm{vdw}_l(t)$. Now for any $S \subseteq \{1,\ldots,n\}$ with $|S| \leq m$ the set $\{1,\ldots,n\} \setminus S$ contains at least one interval $\{i,\ldots,j\}$ for $1 \leq i \leq j \leq n$ with $j-i+1 = \mathrm{vdw}_l(t)$. Using the invariance of linear progressions under translation, we obtain the desired inequality. □

3.2 Arithmetic progressions in the prime numbers

We turn to a major strengthening of Szemerédi's theorem. Now the hypergraph sequence is given as $G_n = \mathrm{ap}_{\mathrm{pr}}(k,n)$ for fixed $k \in \mathbb{N}$, where the vertex set of $\mathrm{ap}_{\mathrm{pr}}(k,n)$ is the set of the first n prime numbers, while the hyperedges are the arithmetic progressions of size k (within the first n prime numbers). As before, every set of prime numbers having at most two elements is an arithmetic progression, but now the first arithmetic progression of size 3 is $\{3,5,7\}$, and the first arithmetic progression of size 4 is $\{5,11,17,23\}$. Until 2004 even condition (i) was unknown, that is, whether the primes contain arbitrarily long arithmetic progressions, and only with [9] not only condition (i) was proven, but even condition (iii) (the underlying preprint was a major contribution towards the Fields medal for Terence Tao in 2006). Actually, until today no other proof of property (i) is known than through property (iii)! In analogy to Definition 1, and as a special case of Definition 6, we define *Green-Tao numbers* ("GT-numbers").

Definition 13. *For a parameter tuple $(k_1,\ldots,k_m)$ let the* ***Green-Tao number*** $\boldsymbol{\mathrm{grt}_m(k_1,\ldots,k_m)}$ *be defined as the smallest $n_0 \in \mathbb{N}$ such that for every $n \geq n_0$ and every $f : \{p_1,\ldots,p_n\} \to \{1,\ldots,m\}$, where $p_1,\ldots,p_n$ are the first n prime numbers, there exists some $i \in \{1,\ldots,m\}$ such that $f^{-1}(i)$ contains an arithmetic progression of size k_i.*

According to Definition 2 we speak of ***trivial GT-numbers****,* ***simple GT-numbers****,* ***core GT-numbers****,* ***transversal GT-numbers****,* ***extended core GT-numbers****, and* ***diagonal GT-numbers****.*

Theorem 10 applied to Green-Tao numbers, using Green-Tao's theorem ([9]), yields that extended GT-numbers grow linearly. For the explicit statement, as for Corollary 11 and as a special case of Definition 7, we use $\mathrm{cr}^{\mathrm{pr}}_{\mathrm{ap}}(t,q)$ for parameter tuples t, using the hypergraph sequences in the primes belonging to the progression sizes in t.

Corollary 14. *For a parameter tuple t of length $l \in \mathbb{N}$, for $m \in \mathbb{N}_0$ and for $s \in \mathbb{R}_{>1}$ we have* $\mathrm{grt}_{m+l}((2,\ldots,2);t) \leq \max(s \cdot m + 1, \mathrm{cr}^{\mathrm{pr}}_{\mathrm{ap}}(t, 1 - \frac{1}{s}))$.

3.3 Remarks on transversal numbers and transversal extensions

Given a nontrivial monotonic hypergraph sequence G, the (computational) determination of the simplest transversal extension numbers $\mathrm{N}_{1+m}(G, Q^2, \ldots, Q^2) = \mathrm{N}_2(G, Q^{m+1})$ (recall Theorem 10), with the special cases $\mathrm{vdw}_{m+1}(2, \ldots, 2, k)$ and $\mathrm{grt}_{m+1}(2, \ldots, 2, k)$, is relatively(!) easy, since essentially we have to compute the *transversal numbers* $\tau(G_n)$ of the hypergraphs G_n (though still an NP-complete task in general), that is the minimum size of a set of vertices having non-empty intersection with every hyperedge. This is also the motivation for the notion of "transversal N-number" and "transversal extension": $\mathrm{N}_2(G, Q^{m+1})$ is the smallest n such that $\tau(G_n) > m$. The complements of independent sets in a hypergraph G are exactly the transversals of G, and thus $\tau(G) + \alpha(G) = |V(G)|$ holds. So determination of the transversal numbers for the hypergraph sequence G determines the convergence rate w.r.t. the Szemerédi property, and is therefore of strong interest (recall Lemma 5).

Considering the computation of $\tau(G_n)$ for the vdW- and the GT-sequence of hypergraphs, going from G_n to G_{n+1} only one vertex is added, and thus we have a relatively slow growth of complexity compared to transversal extensions of core tuples, without a clear boundary of what becomes "infeasible". These problems also require some special treatment (using cardinality constraints or special hypergraph transversal algorithms). So we put the results on transversal vdW- or GT-numbers only into the appendix (see Section A), where we used the most direct method for computing transversal numbers of hypergraphs via SAT (see the introduction to Section A).

- Special methods are applicable regarding the transversal numbers of vdW-hypergraphs, which exploit the translation invariance of arithmetic progressions; see [27,28] for the basic ideas, and see the case $k = 3$ in Subsection A.1 for data derived by such special methods. [22] even found for "small" m a precise formula for $\mathrm{vdw}_{m+1}(2, \ldots, 2, k)$.
- Regarding $\mathrm{grt}_{m+1}(2, \ldots, 2, k)$, such a (simple) formula likely does not exist, and also we loose translation invariance of the arithmetic progressions (since they must lie in the primes), so computing the minimum size of hypergraph transversals via SAT solving seems a good option, but still should (and can) exploit special properties (not investigated in this paper).
- It seems that combining these special methods with SAT solving should yield the best results.

Finally we mention that also transversal extensions of core tuples can be translated into SAT problems by combining the general translation methods of the following section with cardinality constraints (which take care of the initial tuples of 2's). In this article we concentrated on the foundations and on the study of the various general translation schemes, so also the investigations of this special

treatment had to be postponed (that is, the GT-numbers for transversal extensions reported in Section 5 have been obtained by just applying the general translations of non-boolean problems into boolean problems).

4 The generic translation scheme from non-boolean clause-sets to boolean clause-sets

GT-problems of the form "$\mathrm{grt}_2(k_1, k_2) > n$?" have a natural formulation as (boolean) SAT problems by just excluding the arithmetic progressions of sizes k_1 and k_2, e.g. the problem "$\mathrm{grt}_2(2,3) > 4$?" yields the (satisfiable) clause-set $\big\{ \{2,3\}, \{2,5\}, \{2,7\}, \{3,5\}, \{3,7\}, \{5,7\}, \{-3,-5,-7\} \big\}$ over the variable-set $\{2,3,5,7\}$ (thus the answer is "yes"). A natural translation for arbitrary m is given when using *generalised clause-sets* as systematically studied in [17,19,20], which allow variables v with finite domains D_v and literals of the form "$v \neq \varepsilon$" for values $\varepsilon \in D_v$. The problem of colouring a hypergraph G with m colours is naturally translated into a SAT problem for generalised clause-sets via using m clauses for every hyperedge $H \in E(G)$, namely for every value $\varepsilon \in \{1, \dots, m\}$ the clause $\{v \neq \varepsilon : v \in H\}$, stating that not all vertices in H can have value ε (note that the vertices of G are used as variables with (uniform) domain $\{1, \dots, m\}$). Accordingly we arrive at the natural generalisation $\mathrm{F}^{\mathrm{GT}}_{k_1,\dots,k_m}(n)$ of the boolean formulation, using as variables the first n prime numbers, each with domain $\{1, \dots, m\}$, where the clauses are obtained from the hyperedges of $\mathrm{ap}_{\mathrm{pr}}(k_i, n)$ for $i \in \{1, \dots, m\}$ by using literals "$v \neq i$".

As a running example consider $m = 3$, $k_1 = k_2 = k_3 = 3$ and $n = 5$. We remark that we have $\mathrm{grt}_3(3) = 137$, as can be seen in Section 5. Only one hypergraph needs to be considered here (since all k_i-values coincide), namely $\mathrm{ap}_{\mathrm{pr}}(3,5) = (\{2,3,5,7,11\}, \{\{3,5,7\}, \{3,7,11\}\})$. Now the (non-boolean) clause-set $\mathrm{F}^{\mathrm{GT}}_{3,3,3}(5)$ uses the five (formal[6)]) variables $2, 3, 5, 7, 11$, each with domain $\{1,2,3\}$, while we have $3 \cdot 2 = 6$ clauses (each of length 3), namely the clauses $\{(3,i), (5,i), (7,i)\}, \{(3,i), (7,i), (11,i)\}$ for $i \in \{1,2,3\}$.

In [17] the *nested translation* from generalised clause-sets to boolean clause-sets was introduced, while the generalisation to the *generic translation scheme* is outlined in [20]. Given a generalised clause-set F, for every variable an (arbitrary) unsatisfiable boolean clause-set $T(v)$ is chosen, such that for different variables these clause-sets are variable-disjoint. Furthermore for every value $\varepsilon \in D_v$ a necessary clause $\gamma_v(\varepsilon) \in T(v)$ is chosen (that is, $T(v) \setminus \{\gamma_v(\varepsilon)\}$ is satisfiable), such that to different values different clauses are assigned. Now the translation $T_\gamma(F)$ of F under T and γ replaces for every clause $C \in F$ the (non-boolean)

[6)] note that variable 2 does not occur here; it occurs only for $k_i = 2$, and one could ignore it in general, however then we always had to use the offset 1 when comparing with prime number tables

literals $v \neq \varepsilon$ by the (boolean) literals in clause $\gamma_v(\varepsilon)$, and adds for every variable $v \in \mathrm{var}(F)$ the clauses of the (boolean) clause-set $T(v) \setminus \{\gamma_v(\varepsilon) : \varepsilon \in D_v\}$. The clauses $\gamma_v(\varepsilon)$ are called the *main clauses* of $T(v)$, while the other clauses of $T(v)$ constitute the *remainder*.

Lemma 15. *$T_\gamma(F)$ is satisfiability-equivalent to F.*

Proof. If φ is a satisfying assignment for F, then for every variable $v \in \mathrm{var}(\varphi)$ choose a satisfying assignment ψ_v of $T(v) \setminus \{\gamma_v(\varphi(v))\}$, and the union of these (compatible) assignments ψ_v yields a satisfying assignment for $T_\gamma(F)$ (here it is used that for $\varepsilon \in D_v \setminus \{\varphi(v)\}$ we have $\gamma_v(\varepsilon) \neq \gamma_v(\varphi(v))$). If on the other hand ψ is a satisfying (total) assignment for $T_\gamma(F)$, then for every clause-set $T(v)$ there exists some $\varepsilon_v \in D_v$ such that the clause $\gamma_v(\varepsilon)$ is falsified by ψ; now the assignment $v \mapsto \varepsilon_v$ satisfies F. $\square$

The seven instances of the generic scheme used in this paper, where the domain of variable v is $\{1, \dots, m\}$, and where the boolean variables are v_i for appropriate indices i, are as follows:

1. $T(v) = D_m := \big\{ \{v_1\}, \dots, \{v_m\}, \{\overline{v_1}, \dots, \overline{v_m}\} \big\}$ with m variables is used for the *weak direct translation*, where $\gamma_v(i) := \{v_i\}$. D_m is a marginal minimally unsatisfiable clause-set[7] with deficiency 1 (that is, with $m+1$ clauses). The *strong direct translation* uses $T(v) = D'_m := D_m \cup \{\{v_i, v_j\} : 1 \leq i < j \leq m\}$ and the same γ_v.[8]
2. The *weak reduced translation* uses $m-1$ variables with $T(v) = D_{m-1}$ and an arbitrary bijection γ_v (note that D_{m-1} has m clauses), while the *strong reduced translation* uses the same γ_v and $T(v) = D'_{m-1}$. Different from the direct translations, here γ_v plays a role now, namely the question is to which value one associates the long clause $\{\overline{v_1}, \dots, \overline{v_{m-1}}\}$, and so we have m (essentially) different choices.
   Note that clause-set D_{m-1} can be obtained from D_m by DP-reduction for variable v_m (replacing all clauses containing variable v_m by their resolvents on v_m), and accordingly from a clause-set translated by the (weak/strong) direct translation we obtain the clause-set translated by the (weak/strong) reduced translation by performing DP-reduction on all such variables v_m (using that the remainder-clauses are just used as they are, without additional literals in them).

[7] See [13] for an overview on minimally unsatisfiable clause-sets.

[8] In [25] the "strong direct translation" is called "direct encoding", starting from arbitrary CSP-problems (while we start from generalised clause-sets). We prefer to distinguish between "encodings", which are about variables and the mapping of assignments, and "translations", which concern the whole process, and which can use quite different but semantically equivalent clause-sets for example. For the direct translation it seems that always the strong form is better, but this is not the case for other translations, and so we explicitely distinguish between "weak" and "strong".

3. The *weak nested translation* uses $m-1$ variables and $T(v) = H_{m-1}$, where

$$H_m := \{\, \{v_1\}, \{\overline{v_1}, v_2\}, \dots, \{\overline{v_1}, \dots, \overline{v_{m-1}}, v_m\}, \{\overline{v_1}, \dots, \overline{v_m}\} \,\},$$

using some arbitrary bijection γ_v (note that H_m has deficiency 1, and thus H_{m-1} has m clauses). H_m is up to isomorphism the unique saturated minimally unsatisfiable Horn clause-set with m variables, and in fact is a saturation of the minimally unsatisfiable clause-set D_m (see [13]). The *strong nested translation* uses the same γ_v, and, similar to the strong direct translation, $T(v) = H'_{m-1} := H_{m-1} \cup \{\{v_i, v_j\} : 1 \le i < j \le m-1\}$. For both forms now we have $m!/2$ (essentially) different choices for γ_v (note that only the two clauses of length m in H_m can be mapped to each other by an isomorphism of H_m). The motivation for the introduction of the weak nested translation in [17,20] was that first the number of clauses is not changed by the translation, that is, $T(v)$ is minimally unsatisfiable (also D_{m-1} fulfils this), and second that $T(v)$ is a hitting clause-set, that is, every pair of different clauses clashes in at least one variable. These two requirements ensure that the conflict structure of the original (non-boolean) clause-set is preserved by the (boolean) translation. Instead of using H_{m-1} one could actually use any unsatisfiable hitting clause-set with m clauses here.
4. The *simple logarithmic translation*[9)] considers the smallest natural number p with $2^p \ge m$, and sets $T(v) = A_p$, where A_p consists of all 2^p full clauses over variables $v_1, \dots, v_p$, while γ_v is an arbitrary injection.[10)]

With the exception of the direct translation, which is fully symmetric in the clauses $\gamma_v(\varepsilon)$, one has to decide about the choice γ_v of necessary clauses. With the exception of the simple logarithmic translation this is the choice of a suitable bijection, i.e., a question of ordering the values of the variables. In this initial study we have chosen a "standard ordering", with the aim of minimising the size of the clause-set, by simply assigning the larger clauses to the larger k-values (since the larger the size of arithmetic progressions the fewer there are). Considering our running example $\mathrm{F}^{\mathrm{GT}}_{3,3,3}(5)$ we obtain the following 7 translations:

1. For the direct encoding we get $5 \cdot 3 = 15$ boolean variables $v_{p,i}$ for $p \in \{2,3,5,7\}$ and $i \in \{1,2,3\}$. The clause $\{(3,i),(5,i),(7,i)\}$ is replaced by $\{v_{3,i}, v_{5,i}, v_{7,i}\}$ for $i \in \{1,2,3\}$, while clause $\{(3,i),(7,i),(11,i)\}$ is replaced by $\{v_{3,i}, v_{7,i}, v_{11,i}\}$. For the weak translation we have the 5 additional clauses

9) called the "log encoding" (for CSP-problems) in [25]

10) If $2^p = m$, then there is (essentially) only one choice for γ_v, however otherwise the situation is more complicated, and also resolutions are possible between the remaining clauses, shortening these clauses, and these shortened clauses can be used to shorten the main clauses. Therefore we speak of the "simple" translation, and further investigations are needed to find stronger schemes when $m < 2^p$.

$\{\overline{v_{p,1}}, \overline{v_{p,2}}, \overline{v_{p,3}}\}$ for $p \in \{2,3,5,7,11\}$, while for the strong translation additionally we have the $5 \cdot \binom{3}{2} = 15$ binary clauses $\{v_{p,i}, v_{p,j}\}$ for $p \in \{2,3,5,7,11\}$ and $i, j \in \{1,2,3\}$, $i < j$.

2. For the reduced encoding we get $5 \cdot 2 = 10$ boolean variables $v_{p,i}$ for $p \in \{2,3,5,7,11\}$ and $i \in \{1,2\}$. The clause $\{(3,i),(5,i),(7,i)\}$ is replaced by $\{v_{3,i}, v_{5,i}, v_{7,i}\}$ for $i \in \{1,2\}$ resp. by $\{\overline{v_{3,1}}, \overline{v_{3,2}}, \overline{v_{5,1}}, \overline{v_{5,2}}, \overline{v_{7,1}}, \overline{v_{7,2}}\}$ for $i = 3$, while clause $\{(3,i),(7,i),(11,i)\}$ is replaced by $\{v_{3,i}, v_{7,i}, v_{11,i}\}$ for $i \in \{1,2\}$ resp. by $\{\overline{v_{3,1}}, \overline{v_{3,2}}, \overline{v_{7,1}}, \overline{v_{7,2}}, \overline{v_{11,1}}, \overline{v_{11,2}}\}$ for $i = 3$. For the weak translation there are no additional clauses, while for the strong translation we have $5 \cdot \binom{2}{2} = 5$ additional binary clauses $\{v_{p,1}, v_{p,2}\}$ for $p \in \{2,3,5,7,11\}$.
   Note that due to our standardisation scheme the long replacement-clause is uniformly used for $i = 3$, while actually for each of the five (non-boolean) variables $2,3,5,7,11$ one could use a different $i \in \{1,2,3\}$.
3. For the nested encoding we also get $5 \cdot 2 = 10$ boolean variables $v_{p,i}$ for $p \in \{2,3,5,7,11\}$ and $i \in \{1,2\}$. The clause $\{(3,i),(5,i),(7,i)\}$ is replaced for $i = 1,2,3$ by respectively $\{v_{3,1}, v_{5,1}, v_{7,1}\}$, $\{\overline{v_{3,1}}, v_{3,2}, \overline{v_{5,1}}, v_{5,2}, \overline{v_{7,1}}, v_{7,2}\}$, $\{\overline{v_{3,1}}, \overline{v_{3,2}}, \overline{v_{5,1}}, \overline{v_{5,2}}, \overline{v_{7,1}}, \overline{v_{7,2}}\}$, while clause $\{(3,i),(7,i),(11,i)\}$ for $i = 1,2,3$ is replaced by respectively $\{v_{3,1}, v_{7,1}, v_{11,1}\}$, $\{\overline{v_{3,1}}, v_{3,2}, \overline{v_{7,1}}, v_{7,2}, \overline{v_{11,1}}, v_{11,2}\}$, $\{\overline{v_{3,1}}, \overline{v_{3,2}}, \overline{v_{7,1}}, \overline{v_{7,2}}, \overline{v_{11,1}}, \overline{v_{11,2}}\}$. For the weak translation there are no additional clauses, while for the strong translation we have $5 \cdot \binom{2}{2} = 5$ additional binary clauses $\{v_{p,1}, v_{p,2}\}$ for $p \in \{2,3,5,7,11\}$.
   Note (again) that due to our standardisation scheme the order of the three replacement-clauses is fixed for each variable, while for each variable one could use one of the $3! = 6$ possible orders.
4. Finally, for the logarithmic encoding we get (again, but here this is just an exception) $5 \cdot 2 = 10$ boolean variables $v_{p,i}$ for $p \in \{2,3,5,7,11\}$ and $i \in \{1,2\}$. We use the order $A_2 = \{\{v_1, v_2\}, \{\overline{v_1}, v_2\}, \{\overline{v_1}, \overline{v_2}\}, \{v_1, \overline{v_2}\}\}$, where the first three clauses are used for the values $i = 1,2,3$. Then the clause $\{(3,i),(5,i),(7,i)\}$ is replaced for $i = 1,2,3$ by $\{v_{3,1}, v_{3,2}, v_{5,1}, v_{5,2}, v_{7,1}, v_{7,2}\}$, $\{\overline{v_{3,1}}, v_{3,2}, \overline{v_{5,1}}, v_{5,2}, \overline{v_{7,1}}, v_{7,2}\}$, $\{\overline{v_{3,1}}, \overline{v_{3,2}}, \overline{v_{5,1}}, \overline{v_{5,2}}, \overline{v_{7,1}}, \overline{v_{7,2}}\}$ respectively, and $\{(3,i),(7,i),(11,i)\}$ is replaced resp. by $\{v_{3,1}, v_{3,2}, v_{7,1}, v_{7,2}, v_{11,1}, v_{11,2}\}$, $\{\overline{v_{3,1}}, v_{3,2}, \overline{v_{7,1}}, v_{7,2}, \overline{v_{11,1}}, v_{11,2}\}$, $\{\overline{v_{3,1}}, \overline{v_{3,2}}, \overline{v_{7,1}}, \overline{v_{7,2}}, \overline{v_{11,1}}, \overline{v_{11,2}}\}$. Additionally we have the 5 clauses $\{v_{p,1}, \overline{v_{p,2}}\}$ for $p \in \{2,3,5,7,11\}$.

Somewhat surprisingly, in many cases considered in this paper the weak nested translation turned out to be best (from the above 7 translations considered), for all three types of solvers, look-ahead, conflict-driven and local-search solvers (where for the latter an appropriate algorithm has to be chosen). Only for larger number of colours is the logarithmic translation superior (for local search, with various best algorithms; complete solvers were not successful on any of these instances (with larger number of colours)), while in all cases the weak nested translation was superior over the direct translation (weak or strong, for all solver types).

5 Computing Green-Tao numbers

For trivial GT-numbers as with vdW-numbers we have $\mathrm{grt}_m(2) = m+1$. However the simple GT-numbers are non-trivial: $\mathrm{grt}_1(k)$ is the smallest n such that the first n prime numbers contain an arithmetic progression of size k. Only the values for $2 \le k \le 21$ are known, given by the sequence $2, 4, 9, 10, 37, 155, 263, 289, 316, 21'966, 23'060, 58'464, 2'253'121, 9'686'320, 11'015'837, 227'225'515, 755'752'809, 3'466'256'932, 22'009'064'470, 220'525'414'079$.[11] It seems likely that consideration of GT-numbers for core tuples involving $k \ge 11$ is infeasible (since the first 21966 prime numbers need to be considered just to see the *first* progression of size 11).

Considering the generalised clause-sets $F = \mathrm{F}^{\mathrm{GT}}_{k_1,\dots,k_m}(n)$ (recall Section 4), the number $n(F)$ of (formal) variables is n, while the number $c(F)$ of clauses is $\sum_{i=1}^{m} |E(\mathrm{ap}_{\mathrm{pr}}(k_i, n))|$.[12] So to compute the number of clauses in F, we have to compute how many arithmetic progressions of size k there are for a given n. In other words, what can be said about the number $|E(\mathrm{ap}_{\mathrm{pr}}(k, n))|$ of hyperedges in the GT-hypergraphs? Exploiting the famous (unproven) "m-tuples conjecture" of Hardy-Littlewood, various asymptotic formulas (where the quotient with the true value is approaching 1 with n going to infinity) are given in [10]. Translated into our context, where we rank the primes, for arbitrary $N \in \mathbb{N}_0$ the formula (7) from [10] yields, using $x := n \cdot \log n$ (which is an asymptotically precise formula to translate from the rank n to the associate n-th prime number p_n):

$$|E(\mathrm{ap}_{\mathrm{pr}}(k, n))| \sim C_k \cdot \frac{x^2}{(\log x)^k} \cdot (1 + \sum_{i=1}^{N} \frac{a_{k,i}}{(\log x)^i})$$

(for fixed k, proven meanwhile for $k \le 4$; precise formulas for C_k and the a_i are also given in [10]). Just using linear regression to determine C_k and the $a_{k,i}$, using $N = 2$, yields very good approximations over the ranges we are considering.

Solvers used are the algorithms from the `Ubcsat` local-search suite ([31]), `minisat2` ([7]) for conflict-driven solvers (on our instances either `minisat2` was superior or not much worse than all other publicly available conflict-driven solvers, and thus it seems that the optimisations applied to `minisat2` in other solvers don't improve performance on our instances), and `OKsolver-2002` ([15]), `march_pl` ([12]) and `satz` ([24]) for look-ahead solvers. In one (largest) case `survey propagation` ([4]) was successful (with 708206 clauses of length 5). If not stated otherwise, for all non-boolean cases the weak (standard) nested translation is best (considering complete and incomplete solvers), and if not otherwise

[11] This data is available at `http://users.cybercity.dk/~dsl522332/math/aprecords.htm`, in the form of the prime numbers themselves, not their indices (as used by us), and so we needed to rank the prime numbers there.

[12] It seems that for n not much smaller than $\mathrm{grt}_m(k_1, \dots, k_m)$ all variables actually occur with the exception of prime number 2, which occurs iff some $k_i = 2$ exists.

stated, for lower bounds `rnovelty+` is best. Recall that a lower bound stated as "$\geq n$" means that we conjecture that actually equality holds.

We were able to compute five core GT-numbers, for 3 core numbers we have reasonable conjectures, and for 9 core numbers we have hopefully not unreasonable lower bounds. Furthermore we were able to compute 12 extended core GT-numbers, while for 16 cases we have conjectures. Transversal GT-numbers are presented in Section A.2; see Subsection 3.3 for general remarks.

1. 4 binary core GT-numbers $\mathrm{grt}_2(a,b)$ are known:

a \ b	3	4	5	6	7
3	23	79	528	≥ 2072	> 13800
4	-	512	> 4231		
5	-	-	≥ 34309		

For $(5,5)$ we experienced the only case where `survey propagation` was successful (converging for $n < 34309$, diverging for $n \geq 34309$). For the other lower bounds `adaptnovelty+` is best. `OKsolver-2002` is best for $(4,4)$, while for $(3,5)$ `minisat2` is best, followed by `march_pl`.

2. One ternary core GT-number $\mathrm{grt}_3(a,b,c)$ is known:

a,b \ c	3	4	5
3, 3	137	≥ 434	> 1989
3, 4	-	> 1662	> 8300
4, 4	-	> 5044	

For $(3,3,3)$ the logarithmic translation performed best, with `minisat2` fastest, followed by `OKsolver-2002`. For $(3,4,5)$ `rnovelty` performed best.

3. No core GT-number $\mathrm{grt}_4(a,b,c,d)$ is known:

a,b,c \ d	3	4
3, 3, 3	> 384	> 1052
3, 3, 4	-	> 2750

4. Extending $(3,3)$ by m 2's, i.e., the numbers $\mathrm{grt}_{m+2}(2,\dots,2,3,3)$:

m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	23	31	39	41	47	53	55	≥ 60	≥ 62	≥ 67	≥ 71	≥ 73	≥ 82	≥ 83	≥ 86

`minisat2` is the best complete solver here (also for the other (complete) cases below). For the lower bounds the logarithmic translation is best, with `rsaps` except for $m = 13$ where `walksat-tabu` without null-flips is best.

5. Extending $(3,k)$ for $k \geq 4$ by m 2's, i.e., the numbers $\mathrm{grt}_{m+2}(2,\dots,2,3,k)$:

k \ m	0	1	2	3	4	5	6
4	79	117	120	128	136	≥ 142	≥ 151
5	528	581	≥ 582	≥ 610			

For $k = 5$, $m = 2$ `saps` is best, and for $m = 3$ `walksat`. For $k = 4$, $m = 6$ `walksat-tabu` with the logarithmic translation is best.

6. Extending $(4, k)$ by m 2's, i.e., the numbers $\mathrm{grt}_{m+2}(2, \ldots, 2, 4, k)$:

k \ m	0	1	2
4	512	≥ 553	> 588

`sapsnr` is best (for the lower bounds).

7. Extending $(3, 3, k)$ by m 2's, i.e., the numbers $\mathrm{grt}_{m+3}(2, \ldots, 2, 3, 3, k)$:

k \ m	0	1	2
3	137	151	≥ 154
4	≥ 434	≥ 453	> 471

Some final remarks:

1. For vdW-numbers, Conjecture 3 generalised to all core tuples says that in every row of a table of core numbers (that is, in a core tuple one component grows while the others are fixed) we would have growth-rates $O(n^k)$ (where k depends on the row). Now for GT-numbers a first guess is that we have growth-rates $O(\exp(n^k))$.
2. For vdW-numbers, in [23], Research Problem 2.8.6, it is conjectured that $\mathrm{vdw}_2(k, k) \geq \mathrm{vdw}_2(k-1, k+1) \geq \mathrm{vdw}_2(k-2, k+2) \geq \cdots \geq \mathrm{vdw}_2(2, 2k-2)$. The sequences $N(k, k), N(k-1, k+1), \ldots, N(2, 2k+2)$ can for $N = \mathrm{vdw}_2()$ be reasonably evaluated for $2 \leq k \leq 6$, yielding the sequences

$$(3), (9, 7), (35, 22, 11), (178, 73, 46, 15), (1132, ?, 146, 77, 19),$$

which supports the conjecture. For GT-numbers (that is, $N = \mathrm{grt}_2()$) we can reasonably evaluate $2 \leq k \leq 4$, obtaining the sequences

$$(3), (23, 14), (79, 528, 55),$$

and we see that now we have a more complicated behaviour.

6 Open problems and outlook

Regarding the generic translation scheme, further extensive experimentation is needed w.r.t. the problem of ordering the values and of mixing translation

schemes (recall that every variable can be treated on its own). Also further instances of the generic scheme need to considered, starting with refining the logarithmic translation when the number of values is not a power of 2. Of course, finally some form of understanding needs to be established, and we hope that the generic scheme offers a suitable environment for such considerations.

As mentioned in Subsection 3.3, the translation of transversal extension problems into boolean SAT can use cardinality constraints, and this needs to be explored systematically. This includes the special case of transversal extensions of simple tuples, which is basically the hypergraph transversal problem (for these special hypergraphs).

A fundamental problem is to improve performance on *unsatisfiable* instances (of complete solvers). The most promising general approach seems to us to systematically study the optimisation of heuristics as outlined in [18]. Investigating the tree-resolution and full-resolution complexity of these instances should be of great interest; we noticed that especially with the `OKsolver-2002` the search trees show remarkable regularities (of a number-theoretical touch, in a kind of "fractal" way). Exploiting the monotone nature of the hypergraph sequences of vdW- or GT-hypergraphs seems also necessary to reach the next level of vdW- or GT-numbers (regarding core parameter tuples), where some first (sporadic) methods one finds in [14].

In general, it seems to us that instances from Ramsey theory, like vdW-instances or GT-instances as considered in this paper, or like the Ramsey-instances (and there are many other families), provide very good benchmarks for SAT solvers, combining the power of systematic creation as for random instances with various types of "structures", where the interplay between these structures and SAT solving should be of great interest and potential.

References

1. Tanbir Ahmed. Some new van der Waerden numbers and some van der Waerden-type numbers. *INTEGERS: Electronic Journal of Combinatorial Number Theory*, 9:65–76, 2009.
2. Tanbir Ahmed. Two new van der Waerden numbers: w(2;3,17) and w(2;3,18). To appear in INTEGERS: Electronic Journal of Combinatorial Number Theory, 2010.
3. Armin Biere, Marijn J.H. Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, February 2009.
4. A. Braunstein, M. Mézard, and R. Zecchina. Survey propagation: An algorithm for satisfiability. *Random Structures and Algorithms*, 27(2):201–226, March 2005.
5. Tom Brown, Bruce M. Landman, and Aaron Robertson. Bounds on some van der Waerden numbers. *Journal of Combinatorial Theory, Series A*, 115:1304–1309, 2008.

6. Michael R. Dransfield, Lengning Liu, Victor W. Marek, and Miroslaw Truszczyński. Satisfiability and computing van der Waerden numbers. *The Electronic Journal of Combinatorics*, 11(#R41), 2004.
7. Niklas Eén and Niklas Sörensson. An extensible SAT-solver. In Enrico Giunchiglia and Armando Tacchella, editors, *Theory and Applications of Satisfiability Testing 2003*, volume 2919 of *Lecture Notes in Computer Science*, pages 502–518, Berlin, 2004. Springer. ISBN 3-540-20851-8.
8. Paul Erdös and P. Turán. On some sequences of integers. *Journal of the London Mathematical Society*, 11:261–264, 1936.
9. Ben Green and Terence Tao. The primes contain arbitrarily long arithmetic progressions. *Annals of Mathematics*, 167(2):481–547, 2008.
10. Emil Grosswald and Jr. Peter Hagis. Arithmetic progressions consisting only of primes. *Mathematics of Computation*, 33(148):1343–1352, October 1979.
11. P.R. Herwig, M.J.H. Heule, P.M. van Lambalgen, and H. van Maaren. A new method to construct lower bounds for van der Waerden numbers. *The Electronic Journal of Combinatorics*, 14(#R6), 2007.
12. Marijn J.H. Heule. *SMART solving: Tools and techniques for satisfiability solvers.* PhD thesis, Technische Universiteit Delft, 2008. ISBN 978-90-9022877-8.
13. Hans Kleine Büning and Oliver Kullmann. Minimal unsatisfiability and autarkies. In Biere et al. [3], chapter 11, pages 339–401.
14. Michal Kouril and Jerome L. Paul. The van der Waerden number $W(2,6)$ is 1132. *Experimental Mathematics*, 17(1):53–61, 2008.
15. Oliver Kullmann. Investigating the behaviour of a SAT solver on random formulas. Technical Report CSR 23-2002, Swansea University, Computer Science Report Series (available from `http://www-compsci.swan.ac.uk/reports/2002.html`), October 2002. 119 pages.
16. Oliver Kullmann. The `OKlibrary`: Introducing a "holistic" research platform for (generalised) SAT solving. *Studies in Logic*, 2(1):20–53, 2009.
17. Oliver Kullmann. Constraint satisfaction problems in clausal form: Autarkies and minimal unsatisfiability. Technical Report TR 07-055, version 02, Electronic Colloquium on Computational Complexity (ECCC), January 2009.
18. Oliver Kullmann. Fundaments of branching heuristics. In Biere et al. [3], chapter 7, pages 205–244.
19. Oliver Kullmann. Constraint satisfaction problems in clausal form I: Autarkies and deficiency. *Fundamenta Informaticae*, 2010. To appear.
20. Oliver Kullmann. Constraint satisfaction problems in clausal form II: Minimal unsatisfiability and conflict structure. *Fundamenta Informaticae*, 2010. To appear.
21. Oliver Kullmann. Green-Tao numbers and SAT. In Ofer Strichman and Stefan Szeider, editors, *Theory and Applications of Satisfiability Testing - SAT 2010*, Lecture Notes in Computer Science. Springer, 2010.
22. Bruce Landman, Aaron Robertson, and Clay Culver. Some new exact van der Waerden numbers. *INTEGERS: Electronic Journal of Combinatorial Number Theory*, 5(2):1–11, 2005. #A10.
23. Bruce M. Landman and Aaron Robertson. *Ramsey Theory on the Integers*, volume 24 of *Student mathematical library.* American Mathematical Society, 2003. ISBN 0-8218-3199-2.
24. Chu Min Li. A constraint-based approach to narrow search trees for satisfiability. *Information Processing Letters*, 71(2):75–80, 1999.

25. Steven Prestwich. CNF encodings. In Biere et al. [3], chapter 2, pages 75–97.
26. K.F. Roth. On certain sets of integers. *Journal of the London Mathematical Society*, 28:245–252, 1953.
27. Jr. Samuel S. Wagstaff. On sequences of integers with no 4, or no 5 numbers in arithmetical progression. *Mathematics of Computation*, 21(100):695–699, October 1967.
28. Jr. Samuel S. Wagstaff. On k-free sequences of integers. *Mathematics of Computation*, 26(119):767–771, July 1972.
29. E. Szemerédi. On sets of integers containing no four elements in arithmetic progression. *Acta Mathematica Academiae Scientiarum Hungaricae*, 20:89–104, 1969.
30. E. Szemerédi. On sets of integers containing no k elements in arithmetic progression. *Acta Arithmetica*, 27:299–345, 1975.
31. Dave A.D. Tompkins and Holger H. Hoos. UBCSAT: An implementation and experimentation environment for SLS algorithms for SAT and MAX-SAT. In Holger H. Hoos and David G. Mitchell, editors, *Theory and Applications of Satisfiability Testing 2004*, volume 3542 of *Lecture Notes in Computer Science*, pages 306–320, Berlin, 2005. Springer. ISBN 3-540-27829-X.
32. B.L. van der Waerden. Beweis einer Baudetschen Vermutung. *Nieuw Archief voor Wiskunde*, 15:212–216, 1927.
33. Hantao Zhang. Combinatorial designs by SAT solvers. In Biere et al. [3], chapter 17, pages 533–568.

A Transversal numbers

A generic way of computing transversal vdW-numbers and transversal GT-numbers via SAT-solvers is as follows, using $N \in \{\mathrm{vdw}(), \mathrm{grt}()\}$ and respectively $G_k(n) = \mathrm{ap}(k, n)$ or $G_k(n) = \mathrm{ap}_{\mathrm{pr}}(k, n)$.

- Let $\tau_k(n) := \tau(G_k(n))$.
- Recall that $N_{m+1}(2, \dots, 2, k)$ is the smallest n with $\tau_k(n) > m$.
- So we compute the numbers $\tau_k(n)$ for $n = 1, 2, \dots$ as far as we get, and derive from these transversal numbers the transversal N-numbers.
- We start with $\tau_k(1) = 0$ (for $k > 1$), and we know
$$\tau_k(n+1) \in \{\tau_k(n), \tau_k(n) + 1\}.$$
- So for computing $\tau_k(n+1)$ we consider the satisfiability problem
$$\tau_k(n+1) = b \text{ ?}$$
for $b := \tau_k(n)$: If this problem is satisfiable (a transversal for $G_k(n+1)$ of size b exists), then we have $\tau_k(n+1) = b$, while otherwise we have $\tau_k(n+1) = b+1$.
- To formulate the satisfiability problem as a (boolean) CNF-SAT-problem, we consider the vertices of $G_k(n+1)$ as variables $v_1, \dots, v_{n+1}$ and the hyperedges as positive clauses, and we add clauses expressing the cardinality constraint "$v_1 + \cdots + v_{n+1} = b$" (considering $v_i \in \{0, 1\} \subset \mathbb{N}_0$).

The best combination of SAT-solver and cardinality-constraint-translation we found uses `minisat2` and binary addition. Except for $\mathrm{vdw}_{m+1}(2,\ldots,2,k)$ all numbers are computed in this way.

A.1 Transversal vdW-numbers

Now we present the transversal vdW-numbers $\mathrm{vdw}_{m+1}(2,\ldots,2,k)$ we have computed. Numbers in boldface are given by the (precise) formula in [22], while for underlined numbers the formula still holds though the m-value is not in the domain of proven correctness.

1. The known numbers $\mathrm{vdw}_{m+1}(2,\ldots,2,3)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**3**	**6**	<u>7</u>	<u>8</u>	10	12	15	16	17	18
1	19	21	22	23	25	27	28	29	31	33
2	34	35	37	38	39	42	43	44	45	46
3	47	48	49	50	52	53	55	56	57	59
4	60	61	62	64	65	66	67	68	69	70
5	72	73	75	76	77	78	79	80	81	83
6	85	86	87	88	89	90	91	93	94	96
7	97	98	99	101	102	103	105	106	107	108
8	109	110	112	113	115	116	117	118	119	120
9	123	124	125	126	127	128	129	130	131	132
10	133	134	135	136	138	139	140	141	142	143
11	144	146	147	148	149	151	152	153	154	155
12	156	158	159	160	161	162	164	166	167	168
13	170	171	172	173	175	176	177	178	179	180
14	181	182	183	184	185	186	187	188	189	190
15	191	192	193							

Here via SAT solving only up to $\tau(\mathrm{ap}(3,101)) = 74$, $\tau(\mathrm{ap}(3,102)) = 75$ could be computed (yielding $\mathrm{vdw}_{74+1}(2,\ldots,2,3) = 102$), while the data for $m > 74$ has been collected by Jarek Wroblewski at `http://www.math.uni.wroc.pl/~jwr/non-ave.htm`, using special methods. The data is available in the form of an "α-steplist", that is, for index $i = 1,2,3,\ldots$ the smallest $a_i = n \in \mathbb{N}$ with $\alpha(\mathrm{ap}(3,n)) = i$ is given. Jarek Wroblewski conjectures that $a_i \le i^{1.5}$. This data for $i = b \cdot 10 + c$ is given by the following table.

b \ c	1	2	3	4	5	6	7	8	9	10
0	1	2	4	5	9	11	13	14	20	24
1	26	30	32	36	40	41	51	54	58	63
2	71	74	82	84	92	95	100	104	111	114
3	121	122	137	145	150	157	163	165	169	174
4	194									

We would get such sequence of these numbers also directly from the runs of the SAT-solvers (as discussed above, however only as far as we get), by collecting all the n for which a satisfiable instance was obtained (which means that the transversal number didn't change, which is equivalent to the independence numbers making a step (of $+1$)).

2. The known numbers $\mathrm{vdw}_{m+1}(2,\dots,2,4)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**4**	**7**	**11**	12	14	16	18	20	22	24
1	26	29	31	32	35	36	38	39	41	42
2	44	46	47	49	51	52	55	56	57	59
3	61	62	63	65	67	69	71	72	73	75
4	76	78	80							

3. The known numbers $\mathrm{vdw}_{m+1}(2,\dots,2,5)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**5**	**10**	**15**	**20**	<u>21</u>	22	23	26	30	32
1	35	40	45	46	47	48	50	53	55	60
2	65	70	71	72	73	74	75	80	85	90
3	95	96	97	98	99	100	101	102	103	

4. The known numbers $\mathrm{vdw}_{m+1}(2,\dots,2,6)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**6**	**11**	**16**	**21**	**27**	28	30	31	34	38
1	42	43	47	52	53	55	57	60	63	67
2	69	72	77	78	79	81	84			

5. The known numbers $\mathrm{vdw}_{m+1}(2,\dots,2,7)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**7**	**14**	**21**	**28**	**35**	**42**	<u>43</u>	44	45	47
1	49	54	58	62	66	70	77	84	91	92
2	93	94	96	97	99	105	108			

6. The known numbers $\mathrm{vdw}_{m+1}(2,\dots,2,8)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**8**	**15**	**22**	**29**	**36**	**43**	**51**	52	53	55
1	57	60	64	70	73	79	81	86	93	100
2	101	102	103							

7. The known numbers $\mathrm{vdw}_{m+1}(2,\dots,2,9)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**9**	**18**	**25**	**32**	**39**	**46**	$\underline{53}$	58	59	62
1	66	72	74	77	81	87	91	97	102	106
2	110									

8. The known numbers $\mathrm{vdw}_{m+1}(2, \ldots, 2, 10)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**10**	**19**	**29**	$\underline{34}$	$\underline{41}$	$\underline{48}$	$\underline{55}$	62	65	69
1	74	79	85	89	92	96	101	106	110	

9. The known numbers $\mathrm{vdw}_{m+1}(2, \ldots, 2, 11)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**11**	**22**	**33**	**44**	**55**	**66**	**77**	**88**	**99**	**110**
1	$\underline{111}$	112	113	114	116	118	119	121	129	

10. The known numbers $\mathrm{vdw}_{m+1}(2, \ldots, 2, 12)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**12**	**23**	**34**	**45**	**56**	**67**	**78**	**89**	**100**	**111**
1	**123**	124	125	126	127	129	130	133		

11. The known numbers $\mathrm{vdw}_{m+1}(2, \ldots, 2, 13)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**13**	**26**	**39**	**52**	**65**	**78**	**91**	**104**	**117**	**130**
1	**143**	**156**	$\underline{157}$	158	159	160	162	163		

12. The known numbers $\mathrm{vdw}_{m+1}(2, \ldots, 2, 14)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**14**	**27**	**40**	**53**	**66**	**79**	**92**	**105**	**118**	**131**
1	**144**	**157**	**171**	172	173	174	175	176		

13. The known numbers $\mathrm{vdw}_{m+1}(2, \ldots, 2, 15)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**15**	**30**	**43**	**56**	**69**	**82**	**95**	**108**	**121**	$\underline{134}$
1	$\underline{147}$	$\underline{160}$	$\underline{173}$	184	185	186	188	189		

14. The known numbers $\mathrm{vdw}_{m+1}(2, \ldots, 2, 16)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**16**	**31**	**47**	**58**	$\underline{71}$	$\underline{84}$	$\underline{97}$	$\underline{110}$	$\underline{123}$	$\underline{136}$
1	$\underline{149}$	$\underline{162}$	$\underline{175}$	188	197	199	200	202		

15. The known numbers $\mathrm{vdw}_{m+1}(2,\ldots,2,17)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**17**	**34**	**51**	**68**	**85**	**102**	**119**	**136**	**153**	**170**
1	**187**	**204**	**221**	**238**	**255**	**272**	$\underline{273}$	274	275	276
2	277									

16. The known numbers $\mathrm{vdw}_{m+1}(2,\ldots,2,18)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**18**	**35**	**52**	**69**	**86**	**103**	**120**	**137**	**154**	**171**
1	**188**	**205**	**222**	**239**	**256**	**273**	**291**	292	293	294
2	295									

17. The known numbers $\mathrm{vdw}_{m+1}(2,\ldots,2,19)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	**19**	**38**	**57**	**76**	**95**	**114**	**133**	**152**	**171**	**190**
1	**209**	**228**	**247**	**266**	**285**	**304**	**323**	**342**	$\underline{343}$	344
2	345	346								

A.2 Transversal GT-numbers

1. The known numbers $\mathrm{grt}_{m+1}(2,\ldots,2,3)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	4	7	9	13	14	16	18	21	22	23
1	28	29	30	32	33	36	38	39	40	42
2	43	47	48	49	50	52	55	56	57	59
3	61	62	65	68	69	70	71	72	73	75
4	76	78	80	81						

2. The known numbers $\mathrm{grt}_{m+1}(2,\ldots,2,4)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	9	14	17	22	26	32	35	36	37	45
1	46	51	56	58	61	62	71	72	73	78
2	79	83								

3. The known numbers $\mathrm{grt}_{m+1}(2,\ldots,2,5)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	10	31	32	49	58	61	62	78	87	98
1	107	112	121	123	142	143				

4. The known numbers $\mathrm{grt}_{m+1}(2,\ldots,2,6)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	37	55	64	71	90	97	125	152	162	179
1	201	204	211	212	250					

5. The known numbers $\mathrm{grt}_{m+1}(2,\ldots,2,7)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	155	214	228	232	323	396	570	641	715	796
1	827	872	875	1048	1125	1158	1180			

6. The known numbers $\mathrm{grt}_{m+1}(2,\ldots,2,8)$, with $m = a \cdot 10 + b$:

a \ b	0	1	2	3	4	5	6	7	8	9
0	263	349	665	789	1323	1428	1447	1473	1555	1801
1	1881	1935	1979	2117						